Nonlinear Three-Dimensional Trajectory Following: Simulation and Application

George Hines
Control and Dynamical Systems
Caltech

Senior Thesis
Advisors: Profs. Richard Murray (Caltech), Jonathan How (MIT)

June 6, 2008

Contents

List of Figures

Abstract

In light of recent military requirements for unmanned and autonomous vehicles, research into methods of designing *arbitrary three-dimensional trajectories* and controlling aircraft along them has become vital. In this report, we explore two methods of nonlinear control for the purpose of following three-dimensional trajectories and paths. First, prior work on a dynamic feedback linearization exploiting the differential flatness of the ideal airplane is adapted with the intent of implementing it on a physical testbed in MIT's Realtime indoor Autonomous Vehicle test ENvironment (RAVEN), but poor behavior—both in simulation and in hardware—under moderate levels of joint parameter uncertainty thwarted attempts at implementation. Additionally, the differential flatness technique in its pure form follows *trajectories*, which are sometimes inferior intuitively and practically to *paths*. In the context of unmanned air vehicle (UAV) flight in gusty environments, this motivated the extension of prior work on two-dimensional path following to three-dimensions, and simulations are presented in which the fully nonlinear controller derived from differential flatness follows a trajectory that is generated dynamically from a path. The three-dimensional path-following logic is actually implemented in RAVEN, and results are presented that demonstrate good vertical rise time in response to a step input and centimeter accuracy in vertical and lateral tracking. Future directions are proposed.

Chapter 1

Introduction

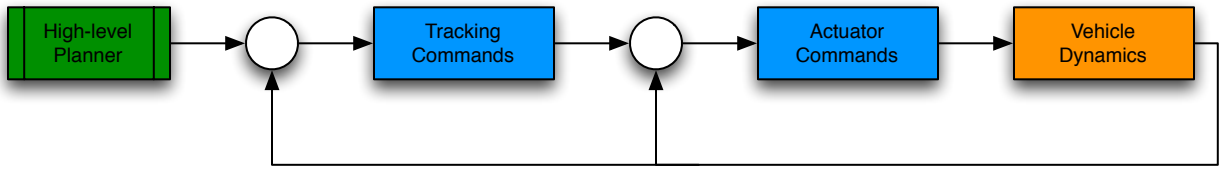
The operation of an aircraft at the extreme edges of its performance envelope continues to require the intervention of a human operator to define critical control inputs. This liability becomes more problematic as aircraft become more maneuverable but, paradoxically, is also more necessary than ever because current trajectory generation schemes cannot keep pace with the capabilities that higher levels of maneuverability provide. The simplest way to overcome the human limitation is to make airplanes *unmanned*, but not necessarily *autonomous*, retaining the human control capabilities by having the pilot command his aircraft from a ground station. Passage of the 2001 National Defense Authorization acknowledged this, requiring one-third of the "operational deep strike force aircraft fleet"[1] to be unmanned by 2010 [1]. The 2007 authorization left this mandate in force, requiring an assessment of progress toward this goal to be included in a Department of Defense policy concerning unmanned systems [2].

However, full autonomy is arriving on the scene, specifically in the form of the RQ-4 Global Hawk [3]. The Global Hawk is a dedicated *reconnaissance* airplane, and therefore is neither required nor expected to perform drastic maneuvers of the kind that are routine for *tactical* aircraft. But it is natural that autonomous—not just unmanned—operation will soon be desired in the operational deep strike force fleet, which counts in its ranks aircraft that cannot be sedate flying platforms. Fully operational air defense systems are multifaceted, so penetration is not guaranteed by pure stealth alone, although this is one example of an advanced capability. To fully stock the quiver of an advanced capability combat airplane, maneuverability is necessary, and it is here that autonomy sees its frontiers in cooperation, task allocation, and path generation/following. In this work we focus on the path following problem.

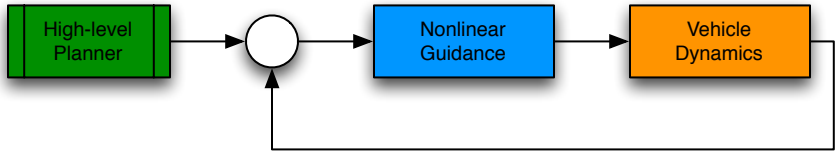
In the methods to be discussed, the existence of a high-level planner that defines a series of goals (in these cases waypoints), which are connected *a priori* by some curve, whether a straight line, a circle, a helix, etc. is assumed. Below this level is a module that determines how the airplane should move to attain that path (this is commonly called the *outer loop*), and within that is the low-level feedback (*inner loop*) that assures that the aircraft carries out the motion prescribed by the outer loop. In such a system, illustrated by the nested feedback structure shown in Figure 1.1(a), the outer loop can be considered

[1] Including the B-2 and F-117 stealth fleet and "at least 30 unmanned advanced capability combat aircraft that are capable of penetrating fully operational enemy air defense systems."

a path generator whose output is tracked by the inner loop, but this makes a distinction that is erased by the nonlinear methods discussed later, which accept as input the high-level path and calculate appropriate actuator commands directly using dynamic feedback. This is visualized in Fig. 1.1(b). It is in the interest of disambiguation that in what follows the path *generator* will refer to the high-level planner and the path *follower* will refer to the (nested) feedback loops that carry out the generator's directives. The methods that we will



(a) Multi-loop control scheme: motion and actuator commands are separate.



(b) Motion and actuator commands are combined.

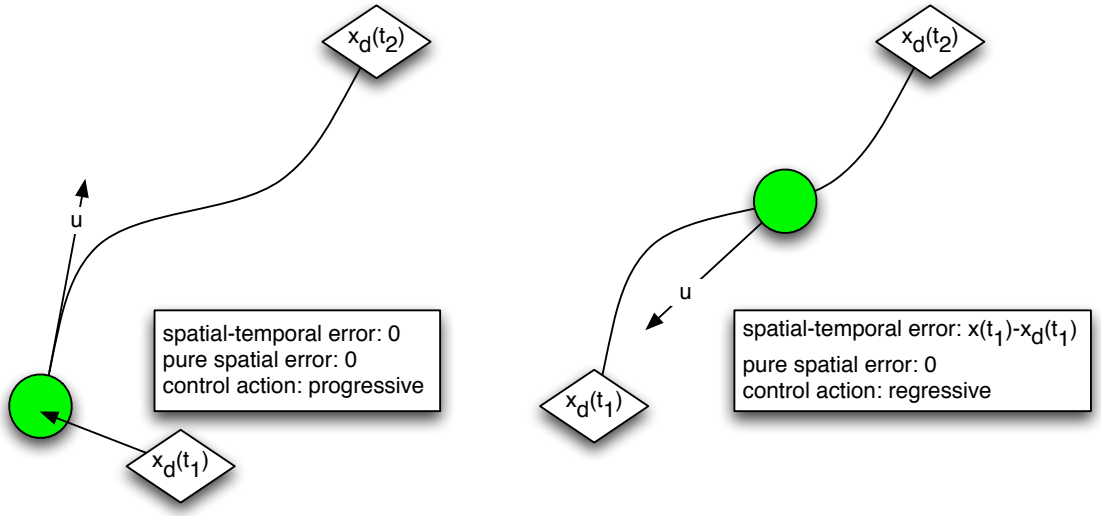
Figure 1.1: Two views of the generator/follower relationship.

discuss are implementations of the "Tracking Commands" block in Fig. 1.1(a) and of the "Nonlinear Guidance" block in Fig. 1.1(b).

Differential flatness allows a full account of the nonlinearities under only mildly restrictive assumptions (such as the neglect of aeroelastic effects). Flatness is well characterized in [4,5], and its application to flight systems is detailed in [6,7]. However, these papers do not address the issue of robustness analytically or in simulation. We will present qualitative studies of the effects of some levels of single parameter uncertainty and joint parameter uncertainty through a closed-loop simulation of a small remote-control airplane.

By nature this particular nonlinear control scheme tracks *trajectories*, which are parameterized by time, and hence often commands large control signals in reaction to spatial-temporal error, even if pure spatial error is identically zero. Consider the situation shown in Figure 1.2. The spatial-temporal and pure spatial errors in 1.2(a) are zero, because the vehicle is at the desired place at the desired time. The control signal is therefore nominal, and moves the vehicle along the path toward the goal, which is taken in this case to be the configuration marked at $t = t_2$. However, the snapshot in 1.2(b) is troublesome. The pure spatial error is zero, because the vehicle is at the correct place for *some* time, in this case $t = t_1 + \tau$. However, for $\tau \neq 0$, this is not *the* correct time, so the spatial-temporal error is nonzero. The resulting control signal will attempt to move the vehicle back to the correct location for $t = t_1$, regressing farther from the goal.

In the occasions where either (a) this behavior is undesirable or (b) it is not critical that the vehicle attain specific locations at specific times (as could easily be the case for a light UAV in a very gusty wind field), this problem motivates the study of *trajectory*



(a) Progressive control signal u, moving the vehicle closer to its goal.

(b) Regressive control signal u, moving the vehicle farther from its goal.

Figure 1.2: The inherent flaw in trajectory following. Both snapshots are taken at time $t = t_1$.

reparameterization, the goal of which is to link the calculation of error to some parameter other than time, which might allow, for example, the situation shown in Figure 1.2(b) to lead to a progressive control signal. Put differently, it is beneficial to specify the desired path so that the controller does not care where on the path the vehicle is, provided it is in fact on the path. Two very different views of this problem are discussed in [8, 9].

A solid base of work had been laid in MIT's Aerospace Controls Laboratory (ACL) concerning a specific type of planar path following method described in [10]: a nonlinear acceleration command generator that will cause the vehicle to fly circular arcs toward the desired trajectory. This algorithm has been proven stable for the straight line following task, and has been experimentally verified to be stable for more general classes of curves, both on full-size outdoor unmanned aerial vehicles (UAVs) and on the small indoor aircraft flown in the Real-time indoor Autonomous Vehicle test ENvironment (RAVEN). It is natural to suppose that a similar algorithm will yield a series of stabilizing acceleration commands in three dimensions, and one such generator has been proven stable in the three-dimensional straight-line following case by Gates [11]. Simulation has indicated good tracking of more general curves, but some steady-state error has been predicted and observed when tracking linearly ascending or descending curves. In what follows we propose an implementation of this three-dimensional nonlinear acceleration command generator, and a characterization of its step response to altitude changes.

Generating acceleration commands can function as a form of trajectory reparameterization. In the case of the controller arrived at through exploitation of differential flatness, the position and its first three derivatives must be specified as functions of time in order to calculate a control signal. Since the path following schemes that I discuss are acceleration command generators, these commands can be integrated twice to obtain a full trajectory

as long as a speed reference is provided and zero jerk is assumed (for convenience). I will attempt to show through simulation that this method of trajectory reparameterization leads to improved (i.e., acceptable) behavior in the presence of spatial-temporal offsets.

In light of the Congressional directive to unman the combat fleet, the motivating context for this research is tactical autonomy. Research is currently underway in the ACL to quantify the highly dynamic air combat environment so that autonomous decisions may be made and carried out. The two-dimensional acceleration generator described above has already been successfully implemented in hardware trials of a simplified planar intercept trajectory generator. As methods of planning combat maneuvers generalize to three dimensions, the three-dimensional following methods described here will be available to smooth the road to further implementation and testing.

Chapter 2

Nonlinear Control with Differential Flatness

2.1 Overview

Traditional flight control systems rely on linearization about multiple trim states, with individual control laws assigned to each through mode switching, gain scheduling, etc. These methods attempt to account for the fundamental nonlinearities of the system by absorbing them into the mode-switching behavior, effectively ignoring their underlying dynamics. We explore instead the property of differential flatness. Several mathematical frameworks have been used to discuss flatness, but let it suffice that an input/output system is flat if there exists "a set of outputs (equal in number to the number of inputs) such that all states and inputs can be determined from these outputs without integration" [4].

The nature of the flat outputs is very sensitive to the formulation of the system. Charlet *et al.* showed that the equations of motion for a six-degree-of-freedom rigid body (of which an airplane is one example) are flat, the outputs being the thrust and the inertial coordinates of the center of mass [5]. This is not the most physically intuitive set of outputs, but using a slightly different description of the state of the system Martin confirmed the airplane's flatness with a set of outputs composed of the sideslip angle and the inertial coordinates of the center of mass [6, 7]. The geometric interpretation of the sideslip angle is considerably more pleasing than that of the thrust, especially when attempting to construct trajectories in the space of flat outputs.

2.2 Control Architecture

Essentially, we implement the control law described by Martin in [6] and summarized in [7], to which readers are directed for the derivation. Changes will be noted as necessary, along with a discussion of applicability to different experimental platforms.

To summarize the main points of Martin's approach: the aircraft's state is defined to be

$$\Xi = (x, y, z, V, \alpha, \beta, \gamma, \chi, \mu)^T,$$

where (x, y, z) are the inertial coordinates of the center of mass, (V, α, β) are the speed, angle of attack and sideslip angle, and (γ, χ, μ) are the Euler angles of the wind axes.

The aircraft response to control input is decomposed into three time scales:

- a slow time scale, which we call the *navigation system*,
- a fast time scale, which we call the *aircraft response*,
- a very fast time scale, which we call the *actuator response*.

The navigation system consists of the equations of motion

$$\begin{aligned}
\dot{x} &= V \cos\chi \cos\gamma \\
\dot{y} &= V \sin\chi \cos\gamma \\
\dot{z} &= -V \sin\gamma \\
\dot{V} &= -g \sin\gamma + \frac{X}{m} \\
\dot{\alpha} &= \frac{g}{V \cos\beta} \cos\gamma \cos\mu - p \cos\alpha \tan\beta + q \\
&\quad - r \sin\alpha \tan\beta + \frac{Z}{mV \cos\beta} \\
\dot{\beta} &= \frac{g}{V} \cos\gamma \sin\mu + p \sin\alpha - r \cos\alpha + \frac{Y}{mV} \\
\dot{\gamma} &= -\frac{g}{V} \cos\gamma - \frac{Y \sin\mu + Z \cos\mu}{mV} \\
\dot{\chi} &= \frac{Y \cos\mu - Z \sin\mu}{mV \cos\gamma} \\
\dot{\mu} &= -\frac{g}{V} \cos\gamma \cos\mu \tan\beta + p \frac{\cos\alpha}{\cos\beta} + r \frac{\sin\alpha}{\cos\beta} \\
&\quad + \frac{Y \cos\mu \tan\gamma}{mV} - \frac{Z(\sin\mu \tan\gamma + \tan\beta)}{mV},
\end{aligned}$$

where the control inputs are the rotational rates (p, q, r) and the thrust F, which enters through the body force components (X, Y, Z).

The aircraft response consists of the dynamical system

$$I\dot{\Omega} = \begin{pmatrix} L \\ M \\ N \end{pmatrix} - \Omega \wedge I\Omega,$$

where the inputs are the actual deflections of the control surfaces $\delta_{l,m,n,\ldots}$, which enter through the moment components (L, M, N). The states are the rotational rates $\Omega = (p, q, r)^T$, which along with the thrust control the navigation system. I denotes the inertia tensor.

While the navigation system is unchanged for different actuator configurations, the aircraft response has as many inputs as the number of independent control surfaces on the aircraft in question. A "conventional" airplane is equipped with three: ailerons, elevator, and rudder. As an alternative example, consider the case of a tailless aircraft equipped

with *elevons*: linked flaps that can move opposite each other to induce roll, or in concert to induce pitch. For the conventional airplane, we observe with some rearrangement and expansion that the aircraft response system is affine in the control inputs, but in cases where there are other than three independent control surfaces, care must be taken to assure that the transformation from control inputs to moment components is invertible. In the case of elevons (2 independent controls) the transformation is *not* invertible because the *a priori* transition matrix is in $\mathbb{R}^{3\times 2}$, although it may be possible to augment the dimension of the control vector so that the transition matrix is in $\mathbb{R}^{3\times 3}$, and hence invertible for all physical applications.

Finally, the actuator response is governed by

$$\dot{\delta}_{l,m,n,\ldots} = A\delta_{l,m,n,\ldots} + \tilde{\delta}_{l,m,n,\ldots},$$

where the inputs are the commanded deflections of the control surfaces $\tilde{\delta}_{l,m,n,\ldots}$. The states are the actual deflections which control the aircraft response. Here A is a negative number or stable matrix, depending on the number of controls.

The time scale decomposition allows us to compute control rates for the navigation system and use the analytical solutions to the other two systems to calculate the actuator commands necessary to attain the desired rates.

Finally, recalling that in this formulation of the equations of motion the differentially flat outputs are (x, y, z, β), we find that there exists a diffeomorphism

$$D : (\Xi, F) \mapsto (x, y, z, \dot{x}, \dot{y}, \dot{z}, \ddot{x}, \ddot{y}, \ddot{z}, \beta)^T.$$

Again, the reader is referred to [6] for details. Further, upon successive differentiation of the inertial coordinates of the center of mass in the navigation system, we find the relationship

$$\begin{pmatrix} x^{(3)} \\ y^{(3)} \\ z^{(3)} \\ \dot{\beta} \end{pmatrix} = \mathcal{A} + \mathcal{B} \cdot \begin{pmatrix} p \\ q \\ r \\ \dot{F} \end{pmatrix}. \tag{2.1}$$

So, given a reference trajectory $(x^*, y^*, z^*, \beta^*)^T$, we can invert (2.1) to find the desired rates Ω^* (provided the determinant of $\mathcal{B}$ is nonzero), from which we calculate the desired actuator deflections. The result for $\dot{F}^*$ is then integrated to find the thrust command.

2.3 Stabilization

Let the transformed state $(x, y, z, \dot{x}, \dot{y}, \dot{z}, \ddot{x}, \ddot{y}, \ddot{z}, \beta)^T$ be denoted by Ξ'. With the control input $u = \left(x^{(3)}, y^{(3)}, z^{(3)}, \dot{\beta}\right)^T$, the system is linear in the transformed state:

$$\dot{\Xi}' = A\Xi' + Bu. \tag{2.2}$$

With full state feedback, it is possible to find a gain matrix K such that the closed-loop system

$$\dot{\Xi}' = (A - BK)\Xi'$$

is stable. LQR gains are used in this study, with weights tuned to block high-frequency actuator commands.

Inverting (2.1), we incorporate this stabilizing gain as an error scaling:

$$\begin{pmatrix} p^* \\ q^* \\ r^* \\ \dot{F}^* \end{pmatrix} = \mathcal{B}^{-1} \left[\begin{pmatrix} x^{(3)*} \\ y^{(3)*} \\ z^{(3)*} \\ \dot{\beta}^* \end{pmatrix} - Ke_{\Xi'} - \mathcal{A} \right]. \tag{2.3}$$

The error $e_{\Xi'}$ is the deviation in position, speed, acceleration, and sideslip angle from the time-parameterized reference trajectory:

$$e_{\Xi'} = \Xi' - \Xi'^*.$$

2.4 Parameter Selection

This control architecture depends upon a reasonably good knowledge of the aerodynamic properties of the vehicle which we seek to control. Specifically, the force and moment coefficients must be known across the applicable flight envelope, which further requires knowledge of the stability derivatives: Martin's simulations used complete mass properties and aerodynamic data for the F-4 Phantom. Our physical platform will be a small styrofoam radio-controlled aircraft for which aero data has not been compiled; we therefore rely on intuition and estimation to select the process parameters.

The inertia tensor is estimated to be

$$I = \begin{pmatrix} 0.00235 & 0 & 0 \\ 0 & 0.00172 & 0 \\ 0 & 0 & .00353 \end{pmatrix}.$$

For the force coefficients, we assume zero sideslip and use

$$\begin{aligned} C_x &= -0.5\alpha^3 + 1.1\alpha^2 + 0.03 \\ C_y &= 0 \\ C_z &= \begin{cases} 2\pi\alpha & \alpha < 0.3 \\ \frac{0.6\pi}{\left(0.3-\frac{\pi}{2}\right)^3}\left(\alpha - \frac{\pi}{2}\right)^3 & \text{otherwise} \end{cases}. \end{aligned}$$

The piecewise model for the lift coefficient takes into account stall dyamics. The standard drag model is a function of the lift coefficient, but this is only valid for small angles of attack; we use a cubic approximation that has the correct shape for $0 < \alpha < \pi/2$. The null side force coefficient is a consequence of the zero sideslip assumption.

For the moment coefficients we use the general linear model given in [6]:

$$\begin{aligned} C_l &= C_{l\beta}\beta + \frac{C_{lp}b}{2V}p + \frac{C_{lr}b}{2V}r + C_{ll}\delta_l + C_{ln}\delta_n \\ C_m &= C_{m0} + C_{m\alpha}\alpha + \frac{C_{m\dot{\alpha}}a}{2V}\dot{\alpha} + \frac{C_{mq}a}{2V}q + C_{mm}\delta_m \\ C_n &= C_{n\beta}\beta + \frac{C_{np}b}{2V}p + \frac{C_{nr}b}{2V}r + C_{nl}\delta_l + C_{nn}\delta_n. \end{aligned}$$

Some of these coefficients may be calculated from the geometric properties of the aircraft, and a, b are reference lengths; see chapters 6-8 of [12] for a thorough treatment. Others may only be measured in a wind tunnel, and since that exercise was not carried out, we considered it sufficient to choose coefficients of the proper order of magnitude for the geometry of our testbed.

2.5 Simulation Results

Simulations took place in MATLAB using the `ode113` solver to propagate the vehicle dynamics and integrate the thrust command. The control inputs are updated asynchronously at 50 Hz. Control surface deflections are saturated at $\pi/4$ radians, and the maximum thrust is set at 6 Newtons (approximately twice the weight of the aircraft). Minimum thrust is taken to be 0.1 N.

Test trajectories demonstrate various flight regimes and characteristics (STOL, conventional, turning, climbing, etc.), and initial conditions are chosen to investigate "step response" behavior.

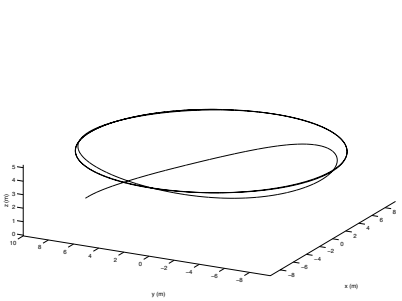
2.5.1 Zero Parameter Uncertainty

Baseline simulations assume perfect parameter knowledge. Subject only to input saturation constraints, the zero-uncertainty proportional-gain system is capable of following modestly complex trajectories in both conventional and STOL flight regimes. However, the saturation constraints substantially restrict the size of the feasible set of trajectories, notably in speed: if the airplane is unable to follow a trajectory in time, it responds drastically to the resulting accumulated error in space. (This is a difficulty inherent in trajectory followers as opposed to path followers.)

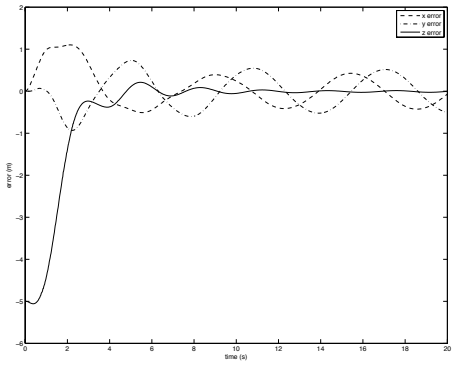
The nominal control signal is proportional, so in certain cases the system admits steady-state error; however, this is currently thought to be a result of the failure of the time scale decomposition, discussed later.

The example presented here is a circle of radius 10 m, nominally flown at a height of 5 m and a speed of 10 m/s. The initial offset is -5 m in elevation. Figure 2.1(a) shows the actual flight path, and it is evident from Figure 2.1(b) that the vehicle does attain the desired height, with no steady-state error. Figure 2.1(b) does show lateral oscillatory error on the order of a meter. This is partly the result of a speed lag that enters with the initial offset. Figure 2.1(c) shows the actual speed, along with the angle of attack, the flight path elevation angle, and the roll angle. The roll angle is stable at about 45 degrees, and the flight path elevation angle exhibits the expected transient increase as the airplane climbs to the desired height. Because of the relatively high speed of this trajectory, the angle of attack remains very small throughout.

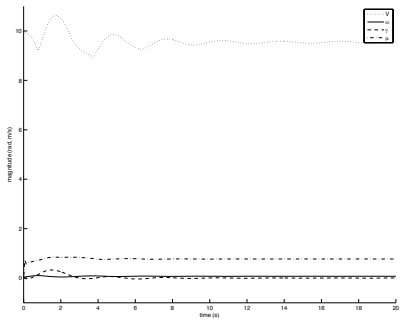
For the remainder of the simulations, the reference trajectory is a straight line along the x-axis flown at 10 m/s. The initial offsets are -2 m laterally, -2 m vertically, and -2 m/s. Discussion will center on the states of the aircraft response system, as this system's stability characteristics dictate the applicability of the time scale decomposition. To establish a reference, Figure 2.2 shows the navigation system (tracking) error and the aircraft response



(a) Flight path.



(b) Tracking error.



(c) Selected states.

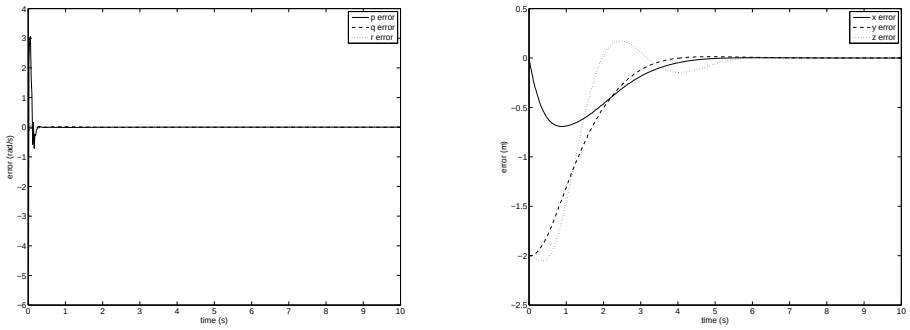
Figure 2.1: Simulation results with perfect parameter knowledge and acceptable parameter values.

system (rate) error for the straight-line trajectory (with initial offsets) with perfect parameter knowledge and acceptable parameter values. Observe that the aircraft response system stabilizes quickly relative to the navigation system, and that the navigation system does not exhibit steady-state error.

(In the subsequent sections, "uncertainty" is modeled as a discrepancy between the parameter value(s) known to the dynamics simulator and the corresponding value(s) known to the controller. The allowable level of uncertainty is taken to be the size of this error at the onset of poor behavior or instability.)

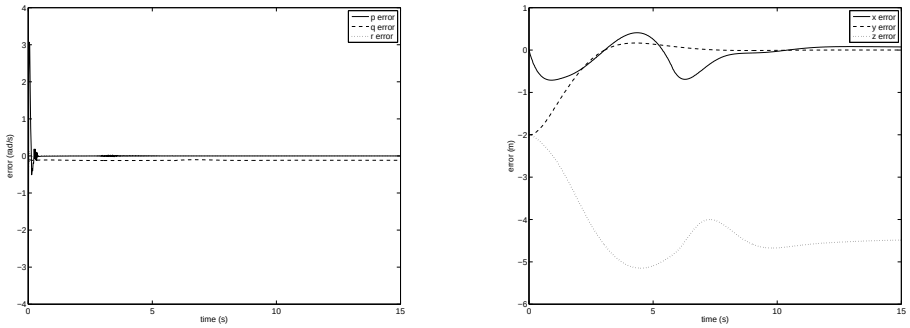
2.5.2 Nonzero Parameter Uncertainty: System Coupling Parameters

The navigation system and the aircraft response system are coupled through the angle of attack and the sideslip angle. Within certain bounds, parameter uncertainty in the coefficients of these coupling terms manifests itself as steady-state error in the navigation system. For example, suppose that the actual aircraft has $C_{m\alpha} = -0.1$ and the controller uses $\widetilde{C}_{m\alpha} = -0.08$. (All other parameters are identical to those that generated Figure 2.2.) With this uncertainty, we obtain the responses shown in Figure 2.3. The aircraft response system, Figure 2.3(a), does stabilize, and quickly, but there is steady-state error in the pitch



(a) Rate error. (b) Tracking error.

Figure 2.2: Tracking and rate errors in response to an initial position and speed offset with perfect parameter knowledge.



(a) Rate error. (b) Tracking error.

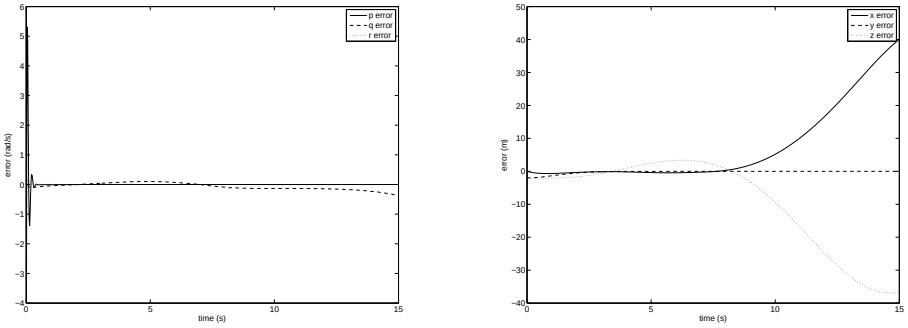
Figure 2.3: Tracking and rate errors in response to an initial position and speed offset with uncertainty in $C_{m\alpha}$.

rate. In the navigation system, Figure 2.3(b), this appears as slow convergence in the lateral coordinates and large steady-state error in height.

Increasing the uncertainty in this parameter quickly leads to divergence, and similar behavior is observed in the coefficient relating yawing moment to sideslip angle. The coefficient relating the rolling moment to the sideslip angle does not exhibit this sensitivity.

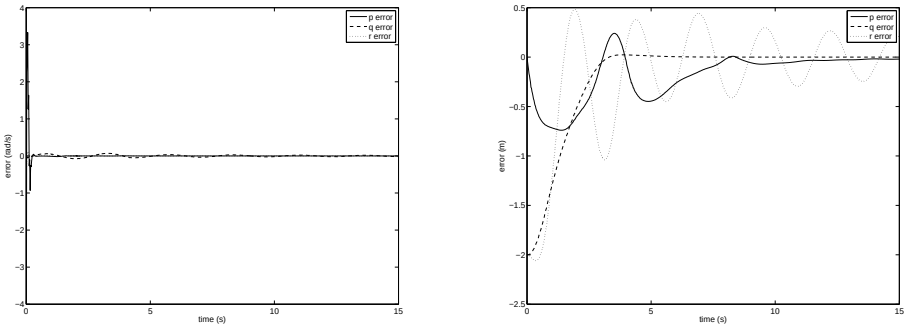
2.5.3 Nonzero Parameter Uncertainty: Decoupled Parameters

The aircraft response system's internal states are scaled by another set of coefficients. The stability of this system is affected relatively little by uncertainty in most of these parameters or their values, provided they are in a reasonable range; the rate derivative C_{mq} is a notable exception. Standard aerodynamic calculations (chapter 7 of [12]) give an approximate value for our platform of $C_{mq} = -22$, but with this value—even with no parameter uncertainty—the aircraft response and navigation systems are unstable. Figure 2.4 shows that the instability in the aircraft response system is slow, but that it is amplified in the



(a) Rate error. (b) Tracking error.

Figure 2.4: Tracking and rate errors in response to an initial position and speed offset with calculated value of $C_{mq} = -22$.



(a) Rate error. (b) Tracking error.

Figure 2.5: Tracking and rate errors in response to an initial position and speed offset with uncertainty in C_{mq}.

navigation system.

In order to get reasonable stability behavior in simulation the absolute value must be reduced by at least an order of magnitude. The previous simulations used the artificial value $C_{mq} = -0.1$, and in order to assess the sensitivity of this parameter to uncertainty, we return to this value and set the controller's knowledge of the parameter to $\widetilde{C}_{mq} = -0.14$. Figure 2.5 shows this case. The slowly-decaying oscillation in the aircraft response system, Figure 2.5(a), invalidates the singular perturbation assumption of fast stability relative to the navigation system, and the result is shown in Figure 2.5(b), with slow stabilization in the lateral coordinates and persistent large-amplitude oscillations in height. Increased uncertainty leads to instability.

2.6 Experimental Results

Implementation of the control law on a physical platform took place in RAVEN, housed at the ACL. Full-state feedback is available from a Vicon MX motion capture system, and multiple vehicles (and vehicle types) may be operated simultaneously.

The platform itself is a styrofoam radio-control airplane built by Ikarus, modeled on the Yak 54 aerobatic monoplane. The wingspan is approximately 0.8 meters, and the nominal thrust-to-weight ratio is close to 2.

Because of the small size of the room, flight testing proved prohibitively dangerous and costly (in terms of demolished airplanes), so initial testing was restricted to taxiing in a tight circle. The airplane exhibited large temporal errors, pausing regularly along the constant-speed trajectory. These temporal errors caused significant spatial errors, which eventually caused the airplane to depart from the desired path.

2.7 Conclusions and Future Work

Simulation has shown that *for an airplane with geometric and aerodynamic parameters similar to those of our testbed, this control architecture is fragile with respect to certain parameter values, and is intolerant of parameter uncertainty.* Specifically, an assumption of the singular perturbation method is that the aircraft response system is asymptotically stable (see chapter 7 of [13] for a complete discussion), but for certain parameter values and for even modest parameter uncertainty, this is not true: the origin is slowly stable or even unstable. Failure of the singular perturbation assumptions invalidates the time-scale decomposition.

It will be instructive to quantify the robustness of this architecture, and this is perhaps best considered alongside the application of adaptive control techniques and system identification to perform online parameter estimation.

In order to apply this control scheme to other projects ongoing both in this lab and elsewhere, it will be convenient to formulate it as a path follower rather than a trajectory follower. Several different methods for trajectory reparameterization have been proposed (for instance [8], which presents a general algorithm), but current focus is on a three-dimensional construction of the guidance logic presented in [10].

Finally, when the architecture is considered functional, it will be natural to extend all results to the multi-vehicle case.

Chapter 3

Three-Dimensional Nonlinear Guidance: Application

3.1 Some Preliminaries

During UAV rendezvous testing at the ACL, a replacement guidance scheme was sought for the existing PD loop closed around the heading angles of the testbed aircraft. The PD controller did not diminish the effects of a steady wind field, and the PID controller designed for that purpose converged more slowly than was desirable. As a solution Park, Deyst, and How proposed the following acceleration command generator in [10]:

$$a_{cmd} = \frac{2v^2}{l} \sin \eta. \tag{3.1}$$

The angle η is the angle between the vehicle's velocity and the desired track. The vehicle's speed is v, and the parameter l tunes the scheme's aggresiveness inversely (larger l corresponds to less aggressive behavior). These measures and their geometric relationships are shown in Figure 3.1. Assuming level flight, the acceleration command thus calculated maps to a unique bank angle, which is subsequently maintained by a PID inner loop. This guidance law is superior to PID control on heading in the particular case of a steady wind field. Further, the linearization of this guidance law yields a simple PD controller, and the classical second-order system parameters (bandwidth and natural frequency) can be determined. Finally, [10] presents a Lyapunov function for the guidance law tracking a straight-line path, completing the analysis of this control scheme in the case of zero-order vehicle dynamics and simple flight regimes. Current work is ongoing to characterize behavior in the presence of first-order dynamics.

3.2 Proposed Three-Dimensional Guidance

Rather than take up the problem of higher-order dynamics, we will focus on extending the guidance law from two spatial dimensions to three. The geometric intuition is identical, but many of the previously scalar quantities must now be considered more generally as vectors.

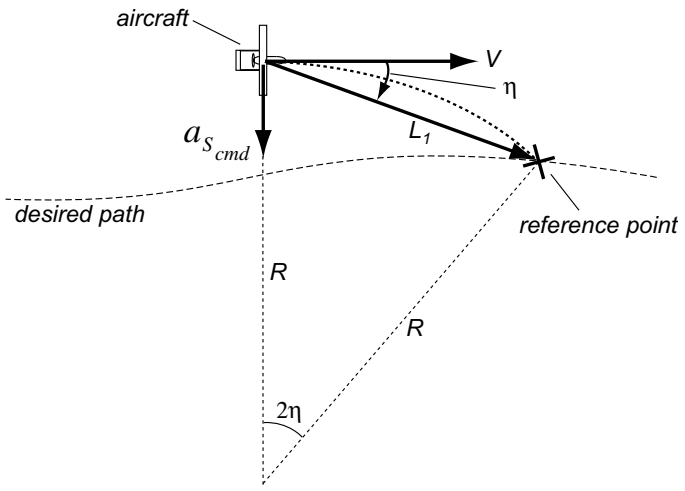


Figure 3.1: The acceleration command a_{cmd} is shown here as $a_{s_{cmd}}$, and will cause the vehicle to follow a circular path of radius R. The vehicle speed, denoted v elsewhere in this paper, is labeled V, and the parameter which this paper refers to as l is called L_1 in this diagram. Taken from [10].

In the three-dimensional formulation, the controller parameter is the same scalar l, but once this parameter is fixed a vector $\mathbf{L}$ is constructed from the vehicle's current position to a point on the desired path such that $|\mathbf{L}| = l$ and the angle between $\mathbf{L}$ and $\mathbf{V}$, the vehicle's vector velocity, is less than $\pi/2$. The existence of such a unique $\mathbf{L}$ is a necessary assumption to apply this guidance strategy.

As in the two-dimensional case, the intuition behind the acceleration command calculation is that we desire an acceleration that will cause a zero-order vehicle to follow a circular course that will nominally intersect the desired path at the endpoint of L. This leads directly to the definition of the acceleration command in vector form:

$$\mathbf{a}_{cmd} = \frac{2}{|\mathbf{L}|^2}(\mathbf{V} \times \mathbf{L}) \times \mathbf{V}. \tag{3.2}$$

The analog to the angle η is implicit in the product $\mathbf{V} \times \mathbf{L}$. The second cross product rotates the acceleration vector into the plane defined by $\mathbf{V}$ and $\mathbf{L}$ such that it is normal to the vehicle's velocity. This fulfills the intuitive goal: the circular path resulting from the application of this acceleration command will lie in the $\mathbf{VL}$ plane, so it will intersect the desired path at the endpoint of $\mathbf{L}$.

3.3 Test Trajectories

The preliminary test objective was to compare the constant-altitude behavior of the three-dimensional guidance scheme to the behavior of the two-dimensional scheme (which is only concerned intrinsically with the constant-altitude case). Therefore the first test trajectory consisted of 16 waypoints approximating a circle at a constant altitude. The waypoints were spaced equally on a circle of radius 2.25 m and connected by straight line segments. (The

problem of calculating three-dimensional accelerations for circular path segments online has not yet been undertaken. The bandwidth of the guidance logic is such that when attempting to follow a 16-edged path inscribed in a circle the resulting flight path is nearly circular.)

Because in the constant-altitude case the tested three-dimensional logic reduces to the two-dimensional method, similar performance is expected. Performance is primarily influenced by the lookahead length: as the lookahead length decreased, the controller aggressiveness increases, and in [10] it is shown that for small deviations from a straight-line path the bandwidth is simply related to the lookahead length. A similar relationship is expected, but has not been verified, for the three-dimensional case. Because performance is tied to the lookahead length parameter, direct comparison of the two-dimensional scheme to the constant-altitude three-dimensional scheme is meaningless unless the effective lookahead lengths are the same.

Due to an implementation detail, making the three-dimensional lookahead length equal to that used in flight tests of the two-dimensional logic was not trivial. The detail is that the test vehicle was started from rest on the ground, and in order to attain the baseline height of the test trajectory, the three-dimensional logic was started at the initial time with a large enough parameter to induce a climb to the desired height. This parameter was necessarily larger than that tested for the two-dimensional case, because in the two-dimensional tests the vehicle was started with an open-loop control sequence to attain the proper height, and then switched to accept commands from the acceleration generator.

As we will soon see, the larger controller parameter significantly impairs tracking, but decreasing the parameter in realtime improves it.

3.4 Test Platform

The vehicle used for flight tests was a small high-wing monoplane constructed of balsa sticks with mylar covering. Its light weight of approximately 20 grams allowed slow level flight, necessary for safety both of the vehicle itself and of the test operators. Instead of the conventional actuator set of ailerons, elevators, and a rudder, this test aircraft was equipped with only elevators and a rudder (roll/yaw coupling induced a deterministic bank angle along with a heading change when the rudder deflected).

Before application, the calculated acceleration command was split into lateral and vertical components, and the simplifying assumption was made that the two components of acceleration were decoupled, with the lateral component executed by a roll maneuver at constant height, and the vertical component executed purely by pitch changes.

3.5 Test Results

For the first flight test the controller parameter was set to 2.5 m. This was suitable to take the vehicle off the ground and up to the desired altitude, but the lateral tracking offsets were on the order of 1 m, which are much larger than those demonstrated by the two-dimensional guidance law. The difference is expected, though, because the available data for the two-dimensional version was captured with a controller parameter of 1.5 m, which leads to much

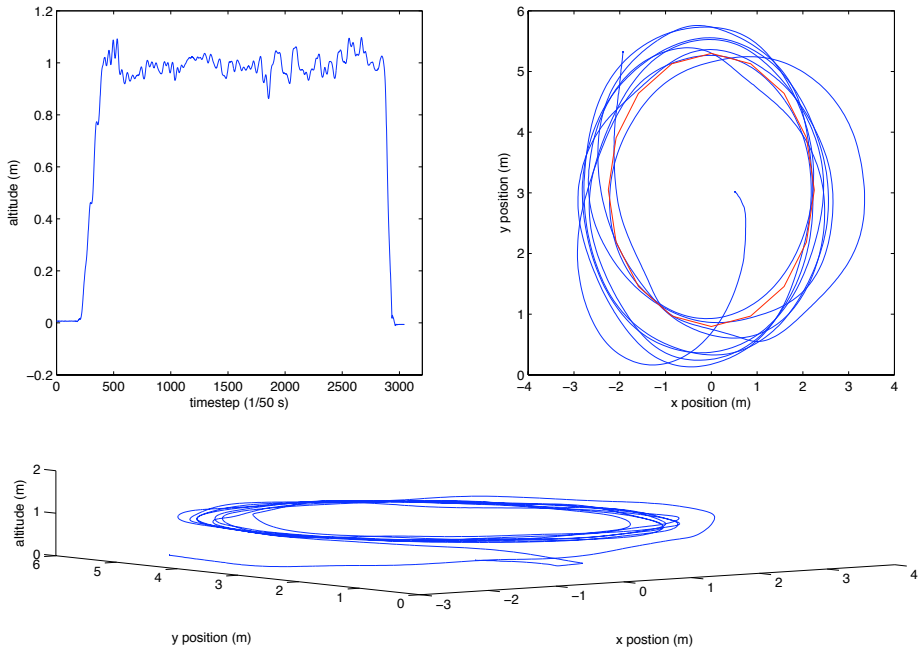


Figure 3.2: Following a "circle" of radius 2.25 m at 1 m altitude with a 2.5 m lookahead length. Altitude tracking is within ±10 cm, but lateral tracking exhibits up to 1 m error. The nominal plan of the flight path is displayed in red on the xy position plot.

more aggressive tracking. The altitude tracking is generally oscillatory within ±10 cm of the reference value.

To resolve this discrepancy while still allowing the vehicle to take off, subsequent test rubrics called for decreasing the controller parameter online after the vehicle had attained the reference height. In the subsequent set of flight tests the lookahead length was decreased incrementally from 2.5 m to 1.5 m after the airplane had taken off and become established on the desired path. The results of this flight are shown in Figure 3.3. Significant improvements are clear in lateral tracking. After the reduction of the lookahead distance, the lateral errors are under 0.5 m, as opposed to consistently approximately 1 m for the previous test. Vertical tracking is essentially unchanged.

The final set of test focused on characterizing the response of the three-dimensional guidance logic to vertical offsets. The commanded path in this case had the same lateral plan (16 waypoints distributed on a circle), but at timestep 1500 a 0.5 m altitude increase was commanded. Because the motor on the test airplane was not sufficiently strong to execute an aggressive climb, it was necessary to leave the lookahead length at 2.5 m when executing flight paths that incorporated altitude variation, so lateral tracking resembles the poorer 1-m bound observed in the first constant-altitude test. The vehicle track is shown in Figure 3.4.

The oscillations about both reference heights are again within about ±10 cm, but there is about 20 cm of overshoot at height changes. Settling time (to normal oscillatory error bounds) is approximately 5 seconds. The oscillatory height errors do not seem to be lessened by decreasing the controller parameter. This may be the result of the somewhat anemic

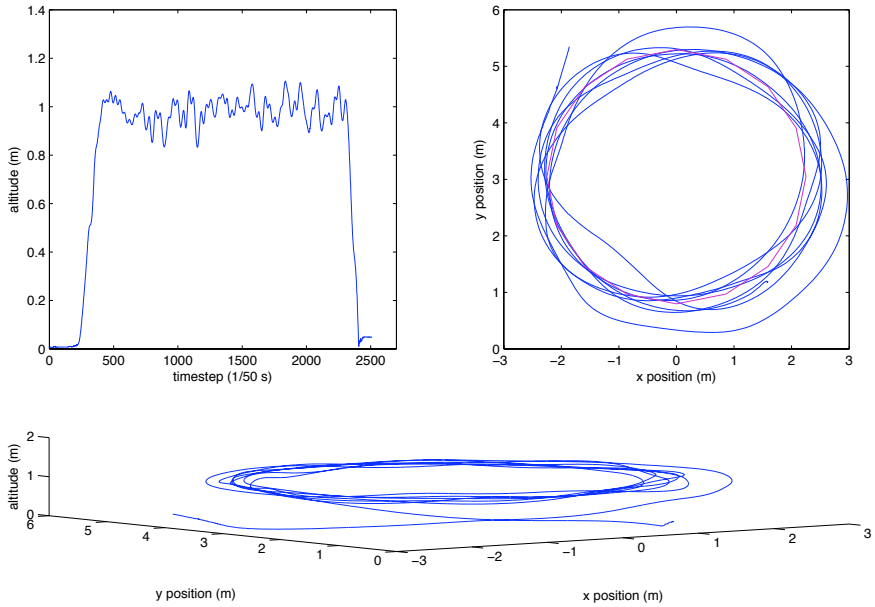


Figure 3.3: Following a "circle" of radius 2.25 m with the lookahead decreasing real-time from 2.5 m to 1.5 m. Lateral tracking is improved.

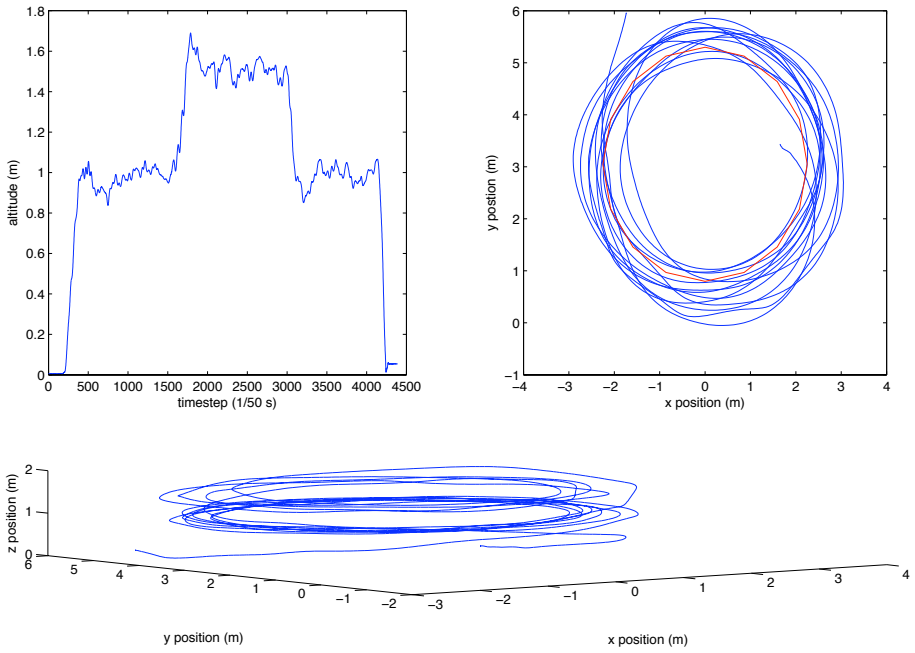


Figure 3.4: Following a "circle" of radius 2.5 m with $|L| = 2.5$ m. At timestep 1500, a 0.5 m step change in altitude is commanded and later reversed.

motor, as it is not capable of the power necessary for quick height changes, and the oscillation may be an artifact of the resulting delay in height response.

3.6 Analysis

The frequency response characteristics of the linearized two-dimensional logic are derived in [10] for small sinusoidal perturbations of a straight-line path at constant altitude; the damping constant is found to be $\zeta = 1/\sqrt{2}$ and the bandwidth is $\omega_0 = \sqrt{2}v/l$, where v is the vehicle speed and l is the lookahead length.

Given a physical system, it is also possible to determine these values empirically (if the system is presumed to be linear and second-order) by measuring the properties of the system's step response. Observing that Figure 3.4 contains such a step response (to the step change in altitude commanded at timestep $T = 1500$), it is possible to deduce the damping constant and bandwidth of the corresponding linear second-order system based on rules laid out in, for instance, [14].

The damping constant can be found directly from the maximum overshoot percentage using the relation $M_p = e^{-\pi\zeta/\sqrt{1-\zeta^2}}$. The maximum percentage overshoot is measured from Figure 3.4 to be close to 16.2%, which gives a damping constant $\zeta \approx 0.5$. With this, there are two possible calculations to determine the bandwidth frequency, involving either the 10%-90% rise time or the 2% settling time. Using the settling time as a metric is not viable in this case because of the oscillatory error present in the altitude tracking. The rise time is much more reliably measured, and is about $T_r = 2.12$ s. With this, we use the relation $T_r = 1.8/\omega_0$, valid for $\zeta = 0.5$, to find the bandwidth $\omega_0 \approx 0.85$.

When the vehicle is following a path with no lateral variation, the vertical acceleration command will follow the same theory as the two-dimensional guidance law. It is, after all, the two-dimensional guidance turned on its side in the case of zero-order dynamics. Therefore, with the results of [10], it is possible to use the vehicle's speed of 3 m/s and lookahead length of 2.5 m to find a predicted bandwidth of $\omega_0 \approx 1.7$ for the situation in which the vehicle is following a path with no lateral variation.

The obvious factor of 2 discrepancy between these values requires some discussion, although we do not have to look far for an explanation. In the case shown in Figure 3.4, the step response in altitude is commanded while the vehicle is flying an approximately circular path, and since the same lookahead vector is used to calculate both lateral and vertical acceleration commands, the aggressiveness of the controller is "split" between these two directions of motion. It comes as no surprise, then, that the bandwidth of the controller in each direction (lateral and vertical) will be less when the controller is excited in both directions simultaneously than when only one direction is perturbed. It is the latter case (perturbation in one direction at a time) that is considered by the analysis in [10].

3.7 Summary

When compared side by side on similar test paths, we have seen that the three-dimensional guidance law does in fact mimic the behavior of the two-dimensional acceleration command

generator, as predicted by the mathematical reduction of the former to the latter in the case of level flight or straight flight. We have also seen that provided the controller parameter is properly matched to the actuator capabilities of the vehicle, good tracking is obtained in response to an altitude step, although empirically the bandwidth is substantially reduced when the vehicle is carrying out lateral and vertical maneuvers simultaneously.

Motivated by the success of this guidance scheme for simple paths and recollection that the nonlinear control scheme presented in the previous chapter requires an acceleration reference as part of its trajectory definition, we now explore the possibility of combining the two techniques by using the acceleration command generator as a seed for the trajectory following capabilities of the controller obtained through feedback linearization.

Chapter 4

Three-Dimensional Nonlinear Guidance: Simulation

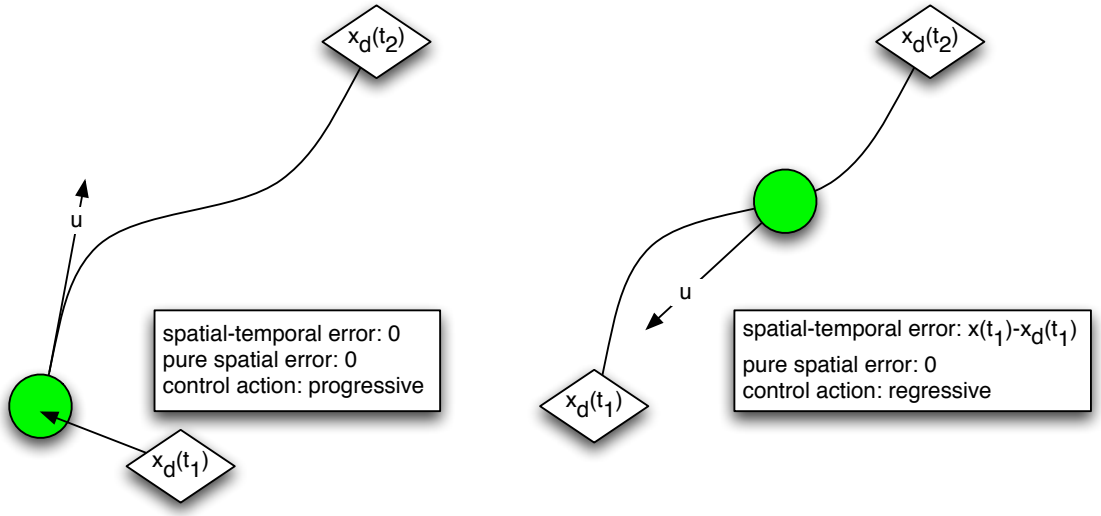
In this section the fully nonlinear control scheme discussed in chapter 2 will be combined with the three-dimensional acceleration command generator. A new form of trajectory reparameterization will be demonstrated based on this acceleration command generator that is suitable to guide a vehicle, using the controller designed with differential flatness, along paths made up of straight line segments.

Trajectory reparameterization is widely studied as a method of meeting actuator constraints. Pappas dealt with this problem extensively in [8]. The motivating problem in his study is an aircraft instructed to be at a specific point in space at a specific time, and in the case that the nominal trajectory requests performance that exceeds the capabilities of the actuator(s), his approach rescales the nominal time evolution until it is within the actuator limits.

An alternative problem is to consider *shifting* the system's time evolution rather than scaling it; consider the situation described in the introduction. (For convenience, the illustration from Chapter 1 is reproduced below.) A formal trajectory is provided which completely specifies the desired time evolution of the vehicle's state at some initial time. However, suppose that it is not necessary for the vehicle to reach the end of that trajectory exactly at the associated terminal time, but rather it is required that the vehicle follow the spatial path described by the formal trajectory.

In the presence of initial temporal offset such as that shown in Figure 4.1(b), the control signal commanded by a pure trajectory-following controller will be regressive. In the pathological case when the desired path is a straight line and the vehicle has a positive temporal offset, the control signal might also be dynamically infeasible; for example, an aircraft may be commanded to slow so drastically that it stalls. Physically, such temporal offsets could occur because of gusting. If a light UAV is flying in a gusty environment, it is conceivable that the UAV could be removed substantially from its nominal trajectory by large gusts, and for the controller to plan dynamically infeasible recovery maneuvers would be unacceptable.

With this alternative goal (good, progressive spatial tracking), the following form of the trajectory reparameterization problem can be posed: given a formal trajectory, guarantee progressive convergence to and following of the spatial path described by the formal trajectory from any initial offset. (The reader is referred to [9] for another view of this problem.)



(a) Progressive control signal, moving the vehicle closer to its goal.

(b) Regressive control signal, moving the vehicle farther from its goal.

Figure 4.1: The inherent flaw in trajectory following. Both snapshots are taken at time $t = t_1$.

The requirement of convergence from *any* initial offset will be relaxed somewhat as the study progresses to requiring convergence from any initial offset that lies within a tube of specified radius around the desired path.

4.1 Formulation

To begin, consider the dynamic feedback compensator derived using differential flatness. Its reference is a trajectory specifying the time evolution of the position and its first three derivatives, and the sideslip angle with its derivative. Denote this trajectory as a continuous function of a reference time τ:

$$\begin{aligned} x_r(\tau) &= f_x(\tau) \\ \beta_r(\tau) &= f_\beta(\tau) \\ \dot{x}_r(\tau) &= f'_x(\tau) \\ \dot{\beta}_r(\tau) &= f'_\beta(\tau) \\ \ddot{x}_r(\tau) &= f''_x(\tau) \\ x_r^{(3)}(\tau) &= f'''_x(\tau). \end{aligned} \tag{4.1}$$

The position x_r is a vector position, and the sideslip β_r is a scalar angle. Assume that zero sideslip is desired, and for convenience that zero jerk is imposed. This latter requirement is

easily relaxed. What remains when these components are ignored is the reduced trajectory

$$\begin{aligned} x_r(\tau) &= f(\tau) \\ \dot{x}_r(\tau) &= f'(\tau) \\ \ddot{x}_r(\tau) &= f''(\tau). \end{aligned} \tag{4.2}$$

Now recall the proposed three-dimensional guidance logic, which generates the lateral acceleration command

$$a_{cmd} = \frac{2}{|L|^2}(V \times L) \times V \tag{4.3}$$

as described in the previous chapter. In order to satisfy the convergence and following constraints, the following set of discrete-time updates is proposed:

$$\begin{aligned} x_r(t_{n+1}) &= x(t_n) + \dot{x}_r(t_{n+1})\delta t \\ \dot{x}_r(t_{n+1}) &= \hat{V}(t_n)v_r + \ddot{x}_r(t_{n+1})\delta t \\ \ddot{x}_r(t_{n+1}) &= \frac{2}{|L|^2}[V(t_n) \times L] \times V(t_n), \end{aligned} \tag{4.4}$$

where $x(t_n)$ is the current position of the vehicle, $V(t_n)$ is the current vector velocity of the vehicle, $\hat{V}(t_n)$ is a unit vector in the direction of $V(t_n)$, v_r is a scalar speed reference, and $\delta t = t_{n+1} - t_n$. Note that even though the formal trajectory is specified as a continuous function, in practice it will be discretized at the controller update rate. The continuous specification is convenient for the present analysis; with it, the issue of asynchronous timesteps when comparing the two specifications does not come up as it would if both were conceived to operate in discrete time.

For the purposes of analysis, assume that the vehicle is a point with zero-order dynamics. It will be shown that over time, the discrete-time reparameterization (4.4) with a suitable speed reference converges to exact tracking of the formal trajectory (4.2) at a shifted time. The demonstration involves two steps. First, *invariance* is proven: if the vehicle is anywhere on the formal trajectory, the proposed discrete reparameterization will keep the vehicle on the formal trajectory for all future time. Next, a lemma is invoked that demonstrates convergence to an arbitrary straight-line path if the proposed acceleration commands are executed.

4.2 Invariance

Let the formal trajectory be completely specified by

$$f(\tau) = l_0 + l_1\tau.$$

Then l_0 is the nominal position at time $\tau = 0$ and l_1 is the desired vector velocity. Now suppose that a vehicle with zero-order dynamics is moving on the trajectory at time τ_1; its position is $l_0 + l_1\tau_1$, its velocity is l_1, and its acceleration is 0. At this time τ_1, the reference specification is switched from (4.2) to (4.4), with the lookahead vector L referenced to the line $f(\tau)$ and the speed reference $v_r = |l_1|$.

Because the vehicle is traveling with the desired velocity and the path is a straight line, $V(t_n) \times L = 0$; moreover, $\hat{V}(t_n) = \hat{l}_1$. It is therefore immediately clear that

$$\begin{aligned} \dot{x}_r(t_{n+1}) &= l_1 \\ \ddot{x}_r(t_{n+1}) &= 0. \end{aligned}$$

Now introduce a specific controller time sequence in which the continuous reference time τ_1 corresponds to controller timestep t_N; then $x(t_N) = f(\tau_1)$. Since the discrete-time velocity and acceleration commands are constant, for every future timestep $n \geq N$ the discrete-time reparameterized position reference will be

$$\begin{aligned} x_r(t_{n+1}) &= x(t_n) + l_1 \delta t \\ &= f(\tau_1) + l_1(t_n - t_N) + l_1 \delta t \\ &= l_0 + l_1(\tau_1 + t_{n+1} - t_N) \\ &= f(\tau_1 + t_{n+1} - t_N). \end{aligned}$$

This is exactly a discretized, timeshifted representation of the formal trajectory reference. In fact, for the straight-line case, (4.4) may be rewritten as

$$\begin{aligned} x_r(t_{n+1}) &= f(\tau_1 + t_{n+1} - t_N) \\ \dot{x}_r(t_{n+1}) &= f'(\tau_1 + t_{n+1} - t_N) \\ \ddot{x}_r(t_{n+1}) &= f''(\tau_1 + t_{n+1} - t_N) \end{aligned} \tag{4.5}$$

for $n \geq N$, provided the state $(x(t_N), \dot{x}(t_N), \ddot{x}(t_N))$ is on the formal trajectory. This demonstrates the spatial invariance of the formal trajectory under the proposed reparameterization scheme.

4.3 Convergence

Gates, in [11], has demonstrated the existence of a positive definite Lyapunov function for the proposed acceleration command generator referenced to an arbitrary straight-line path in three dimensions. So if a vehicle with zero-order dynamics carries out the acceleration commands referenced to some straight line, it will converge asymptotically to that line. (Subject to the requirement that the initial offset cannot be more than $|L|$, in which case the acceleration command is undefined.)

Because of the specification chosen in (4.4), for a given velocity following these commands is equivalent to following the acceleration commands alone and controlling the velocity independently. Therefore using the guidance commands of (4.4), a straight-line formal trajctory is spatially invariant and attracting.

4.4 Simulation Results: Comparison

The motivation for this study was not without practical foundation. While conducting simulations of the dynamic feedback linearization control scheme in response to various

initial offsets, it was observed that offsets along the desired direction of flight resulted in larger control signals and poor dynamic behavior when compared to offsets transverse to the desired direction of flight. It was after realizing that this was due to the time-sensitivity of the formal trajectory that work was begun to smooth the convergence properties so that acceptable forward progress would be maintained regardless of the initial offset.

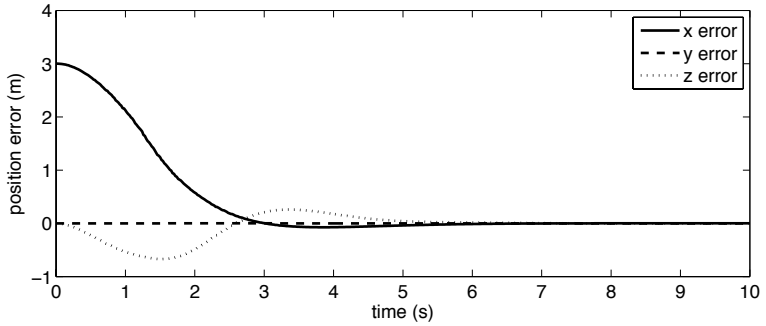
Consider the case when the vehicle's initial condition is on the path described by the formal trajectory at a farther point than that associated with time $\tau = 0$. In the case of the feedback linearized flight controller, the control action will be to slow the aircraft, allowing the trajectory to "catch up" to it, and then speed the aircraft up to bring it to spatio-temporal synchronization. This is demonstrated in Figure 4.2 for a straight-line trajectory along the x-axis to be flown at constant speed 8 m/s, with the aircraft's initial position set at 3 m along the trajectory and initial speed set in the proper direction at the desired speed of 8 m/s. (The relatively short distances and low speeds are a consequence of the nature of the simulated platform, a styrofoam aerobatic remote-control airplane with a wingspan of under 1 m.) To put this in the context of the motivational discussion, it is easy to imagine a situation in which a gusty tailwind displaces a small UAV along its flight path, causing a trajectory-tracking controller to react by slowing the aircraft.

For three meters offset, this behavior is not wholly unacceptable; the vehicle does not approach stall speed, and good control is maintained throughout. However, the larger the arc offset becomes, the poorer the recovering performance. At a 9-m arc offset the vehicle stabilizes extremely slowly, and at a 10-m arc offset the vehicle stabilizes slowly upside down. Meanwhile the actuator signals grow in magnitude and frequency. The evolution of an initial condition with 11 meters of arc offset is shown in Figure 4.3. In this case, the commands are nearly dynamically infeasible, and the moment actuator control signals (shown in Figure 4.4) are either saturated or infeasible. The infeasibility arises from the high-frequency oscillation of the signals, and the elevator signal is also saturated during its ringing pulses.

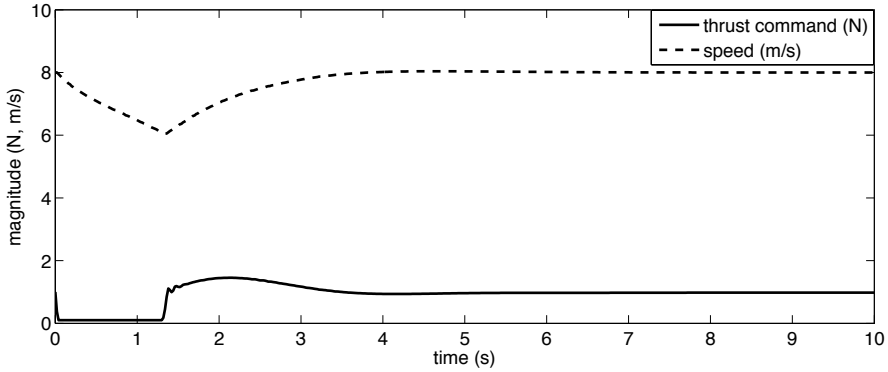
Now consider the same initial condition, but with the control commands generated using (4.4) referenced to the same straight line described by the trajectory used in the previous two examples. The cross-track (y and z) error is on the order of centimeters, Figure 4.5(a), so the vehicle is kept on the desired path without altitude loss as a result of speed reduction as in the previous two examples. The slow oscillations in the z error and the thrust command are both a result of the controller's attempt to maintain a constant speed of 8 m/s. A decrease in thrust command requires a descent if the vehicle is to maintain speed, and correspondingly an increase in thrust must be paired with a climb. Since this trajectory representation does not contain a notion of arc error, it is meaningless to discuss initial arc offset.

4.5 Simulation Results: Capabilities

A more intriguing capability of this reparameterized representation is brought to light when attempting to follow an infeasible path. Figure 4.6 shows the simulation results of following a closed, steeply pitched square path at a commanded speed of 8 m/s. Because of the pitch of the descending segment of the path, this speed command is infeasible; even with the engine producing minimum thrust, the glide angle necessary to follow the commanded straight line will produce a speed greater than 8 m/s. The entire circuit is infeasible in another sense



(a) Position errors.

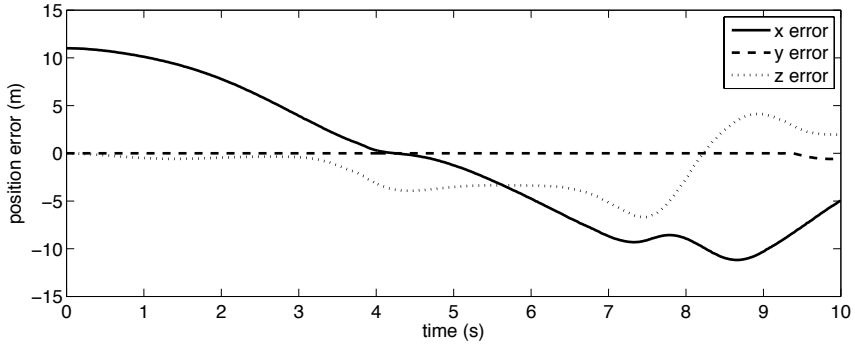


(b) Speed and commanded thrust.

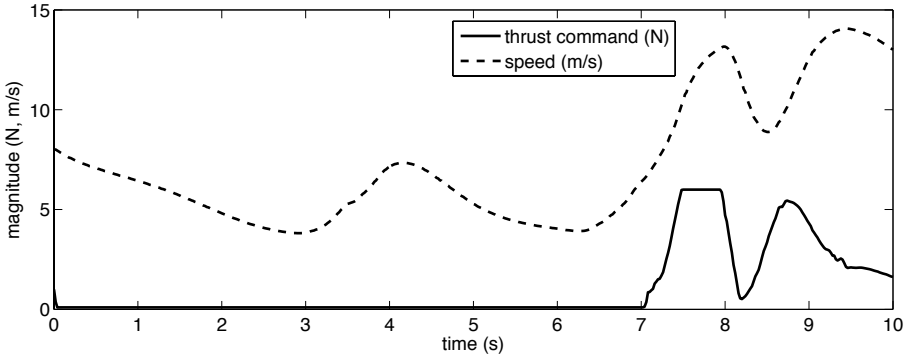
Figure 4.2: The vehicle is started from a point 3 m along the desired trajectory. Its error is referenced to the current time, so although the vehicle is exactly on the desired path (the x-axis), its x error is still nonzero because it is too far along the path (a). The controller's response is to slow from the nominal speed, which is clearly demonstrated in (b) as the speed decreases approximately linearly with the throttle at its lower saturation bound. This speed decrease causes the expected altitude loss, and in (a) it is clear that the z error increases for a time. Around $t = 1.3$ seconds, the trajectory catches up to the slowed vehicle, and the controller begins increasing the vehicle's speed, which brings the vehicle back to the nominal trajectory in both arc position and height.

as well: it is dynamically impossible for a non-holonomic vehicle to negotiate sharp corners. With a pure trajectory follower (which does not incorporate lookahead), a sharp corner will induce overshoot *after* the vehicle passes it, because only then will error be introduced. With a lookahead control design such as the proposed reparameterization, the vehicle will lead each corner, and the overshoot is expected to be small.

There is a trade between tracking performance and acceptable aggressiveness as the controller parameter $|L|$ is varied. For small values of $|L|$ (less than about 5 for the simulated vehicle geometry), tracking is tighter, but the control signals are much larger; short-period oscillations develop as the vehicle reacts to its own drastic actions. For larger values of $|L|$, convergence is slower, but the control signals are much smaller, and long-time tracking is smoother. In Figure 4.6 the controller parameter has been chosen to be $|L| = 10$, which is one quarter the length of each level segment of the path.



(a) Position errors.



(b) Speed and commanded thrust.

Figure 4.3: Here the initial offset of the vehicle is 11 m along the arc of the trajectory. The immediate response of the controller is the same: it cuts the thrust command and waits for the trajectory to "catch up." Here, however, this takes so long that the vehicle loses substantial altitude (which accounts for the increase in speed with no throttle input). Then it must apply maximum thrust to overcome this altitude and speed loss.

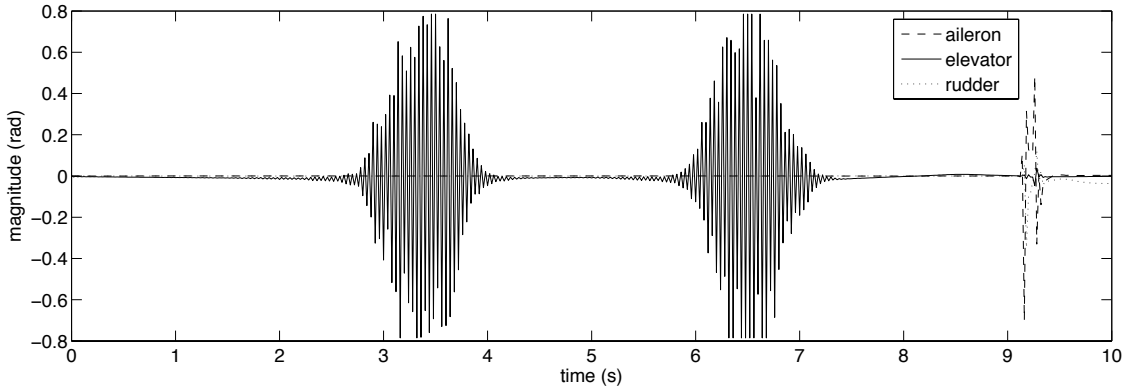
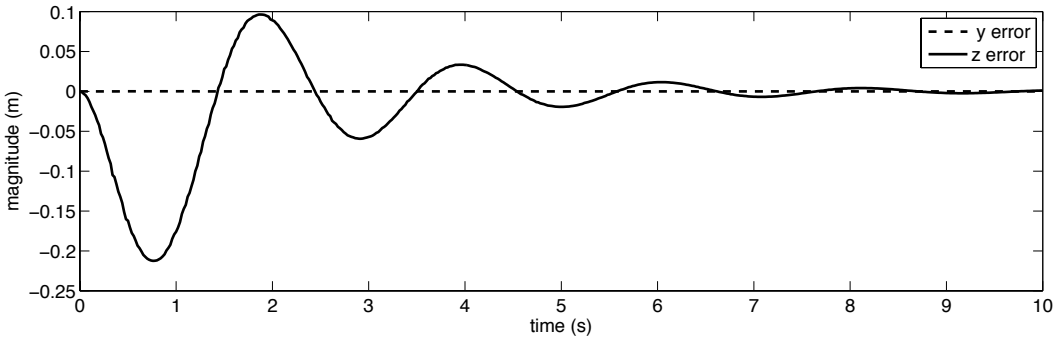
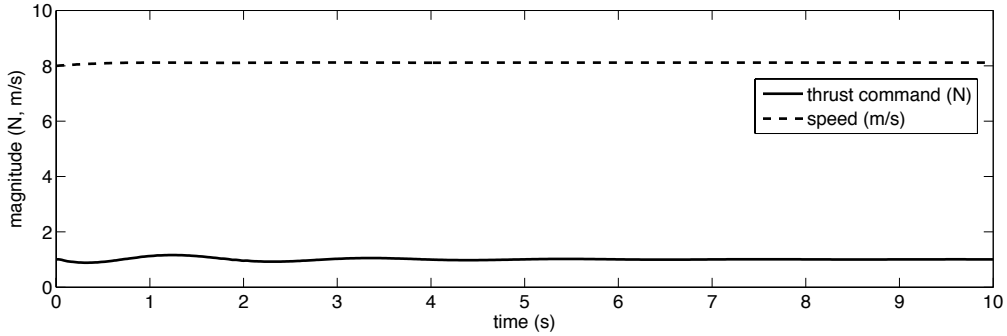


Figure 4.4: During the ringing pulses, the elevator control signal is saturated at $\pi/4$ rad.

The initial condition in this example is offset 3 m vertically and horizontally from the desired path, which begins as a straight line along the x-axis. This diagonally offset initial



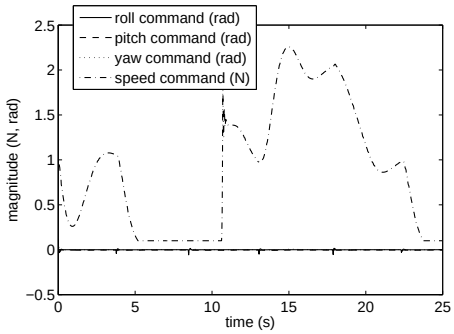
(a) Position errors.



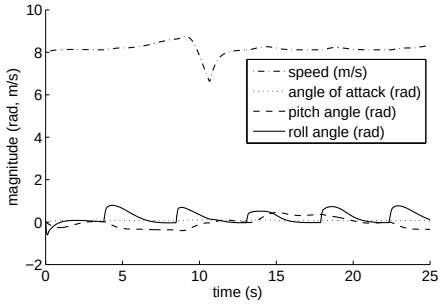
(b) Speed and commanded thrust.

Figure 4.5: When the guidance commands are generated by the acceleration-based reparameterization, the cross-track y error is kept near zero at all times, and the cross-track z error quickly decays to the order of millimeters (a). Since the reparameterized representation does not carry any notion of arc error and since the desired path is along the x-axis, error in the x direction is irrelavent. The speed is not decreased at any point by command (b), because the reparameterized trajectory is always progressive: it always moves the vehicle toward the end of the path.

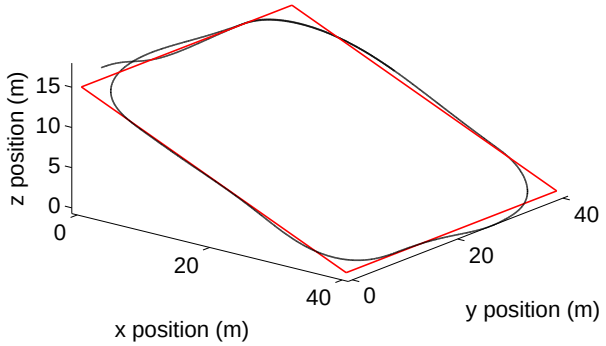
condition highlights the convergence properties of the acceleration guidance method. The second path segment is a steep descent during which the glide speed of the vehicle exceeds the commanded speed. It is clear, however, from the good spatial tracking that this combination of reparameterized guidance commands and the nonlinear controller prioritizes position tracking over speed tracking when the speed command is infeasible. When the vehicle turns into the third path segment, the speed bleeds quickly as the vehicle pulls out of its glide, resulting in a dramatic increase in the thrust command which is quickly damped. As the vehicle turns upward into the fourth segment, the thrust command must increase dramatically to maintain the commanded speed, but this thrust is quickly reduced as the vehicle turns into the upper flat section again, and very little time elapses at this lower thrust setting before the descending section is again reached, the thrust command is cut, and the cycle begins to repeat.



(a) Actuator commands.



(b) Selected states.



(c) Flight path; the commanded path is shown in red, the flight path is shown in black.

Figure 4.6: The vehicle is simulated flying along a pitched, cornered, closed circuit. In (a) are shown the control signals commanded in response to this technically infeasible path. There is one high-frequency pulse in the thrust command, but it is damped quickly. Each pulse in the moment actuator commands corresponds to the start of a cornering turn, and their even spacing implies roughly constant speed throughout. The four most instructive state traces are shown in (b). The hiccup in the speed corresponds to the second turn, from the descending segment into the lower level segment. Each increase in bank angle is a successive right turn. In (c) the vehicle's simulated flight path is overlaid on the nominal path. As expected, the lookahead guidance logic leads the turns with small overshoot of around two meters on a 40 meter path segment. The trajectory ends after completing a full circuit and turning into the descending segment a second time; until the simulation ends, the simulated path through the final turn in the second circuit is within centimeters of the same path in the first circuit.

4.6 Summary

Motivated by consideration of the perils of a light UAV in a gusty wind field, a simple guidance method has been presented to compute control commands at each timestep that are suitable to follow straight-line paths in three spatial dimensions. The benefits of this approach over traditional pure trajectory following are that

- it maintains forward progress regardless of initial arc offset,
- its lookahead action mitigates overshoot at corners,
- it overcomes certain path infeasibilities.

Of course, when the benefits of pure trajectory following are vital (if it is critical that a vehicle be at a certain place at a certain time), it is not appropriate to implement this representation. But in many cases it is acceptable to minimize cross-track error without concern for arc error, and in those cases it is computationally cheap and intuitive to implement the scheme discussed here.

It remains to extend the claims proven here for straight lines to circular paths. If convergence and invariance can be shown for circular paths, then it will be possible to connect any sequence of waypoints in three dimensions with a continuous and differentiable path made up of straight lines and circular arcs, and asymptotic convergence will be guaranteed for each segment.

Chapter 5

Summary and Future Directions

Several aspects of nonlinear flight control and trajectory reparameterization have been discussed, simulated, and implemented on a hardware platform. Within the motivating context of a small autonomous air vehicle in a gusty environment, the stage is set for implementing the methods described previously as the inner loop to a higher-level decision-making planner. There is an extensive foundation that needs to be laid before this integrated architecture can be realized; here we present some avenues of future work that will hasten progress to that end.

5.1 Nonlinear Simulation

The failure of attempts to implement the feedback linearization control scheme on a hardware platform for even the most basic of maneuvers indicates that the method is highly sensitive to parameter uncertainty. A first step in obtaining a metric of this sensitivity is to simulate various levels of single- and joint-parameter uncertainty to obtain a quantitative sketch of the allowable parameter space. However, since the space is high-dimensional this manual brute-force approach is undesirable.

Ideally, a robust control theory for highly nonlinear systems can be developed to analyze such problems rigorously. While this may be years away from becoming a reality, it is possible that parameter cutting techniques can be applied alongside system identification tools to find good enough parameter estimates for practical purposes.

5.2 Three-Dimensional Guidance

Further analysis and testing are necessary before the proposed three-dimensional acceleration-based guidance logic is properly understood. Two important issues are provable stability to arbitrary circular arcs (rather than just arbitrary straight lines) and characterization of performance in the presence of perturbations. The latter performance study would need to characterize the decoupled response (vertical response in straight-line flight or lateral response at constant altitude) as well as the trade that occurs when the vehicle responds to perturbations in both directions simultaneously (the case studied in this report).

5.2.1 Application

It is easy to construct flight test rubrics that isolate the vertical response in straight-line flight from the lateral response in level flight. It is difficult to find test spaces that easily accommodate such tests, because they require relatively long straight-line flights which require a space that is large in at least one direction. An alternative to a large test space is a slow vehicle, and it is very possible that testing with helicopters is more viable than testing with aircraft.

The test trajectories that were flown in these tests are very basic. Steps and straight lines are of course necessary building blocks, but the purpose of designing an intrinsically three-dimensional guidance scheme is to follow more complicated and realistic three-dimensional paths. Because any points in three-dimensional space can be connected by a C^1 path composed of straight lines and circular arcs, the immediate next step is to formulate the guidance law such that it can follow arbitrary circular arcs. This is essentially the problem of finding the vector of specified norm from a point to a circular arc, assuming such a vector exists.

Subsequently it remains to test tracking of more and more complicated paths, specifically in the vicinity of the classical sigularity at 90 degrees pitch and in the situation where the aircraft may need to fly upside down. (Inverted flight carries with it a whole class of problems specific to the state feedback system in the ACL's RAVEN. Solutions are currently being implemented.) The prototype problem is a circular loop flown with a continuous control architecture. Note that for a vehicle with true zero-order dynamics the proposed three-dimensional scheme *will* generate acceleration commands suitable to follow a vertical circle. It is the presence of real vehicle dynamics that complicates the problem.

5.2.2 Analysis and Simulation

The mating of the acceleration command generator and the nonlinear trajectory follower may hold the key to the inverted flight problem (given proper state feedback), although flight in the vicinity of the singularity is still difficult and will require more careful analysis.

There are two open issues that are direct extensions of the work presented here. First, we conjecture that for zero-order dynamics any circular arc in three dimensions is an attractor for the proposed three-dimensional guidance law referenced to that arc. This is intuitively the case given the geometric motivation of the scheme, but it remains to explore and ideally prove this fact. Second, the ideas of trajectory time-shifting presented in chapter 4 need to be extended to the case of circular arcs. This proposed problem is characterized by a steady-state acceleration command in order to follow a circle.

Finally, it is natural to study the effects of higher-order dynamics on the response of a vehicle to the guidance law in three dimensions. This will be an extension of the work currently ongoing to characterize the two-dimensional guidance scheme in the presence of time delays.

Acknowledgments

I would like to thank Profs. Richard Murray and Jon How for their pointed questions and guiding insights during the course of this work. Prof. Philippe Martin graciously provided PDF copies of his dissertation and some papers regarding differential flatness and control in the vicinity of singularities. Prof. John Deyst motivated and encouraged much of the exploration of the three-dimensional guidance scheme. Jim McGrew of the ACL was most helpful in conducting flight tests and integrating the three-dimensional guidance calculations into the existing code structure. Others of the ACL who provided feedback and insight are Brett Bethke, Spencer Ahrens, and Buddy Michini. Eli Cohen was invaluable as he fabricated and helped maintain many of the test aircraft. Thanks also to Dave Rosen for assistance in structuring and editing this report.

Bibliography

[1] 106th U.S. Congress, "National defense authorization, fiscal year 2001," Public Law 106-398, October 2000.

[2] 109th U.S. Congress, "National defense authorization, fiscal year 2007," Public Law 109-364, October 2006.

[3] J. Drezner and R. Leonard, *Innovative development: Global Hawk and DarkStar–flight test in the HAE UAV ACTD program.* Santa Monica, CA: RAND, 2002.

[4] P. Martin, R. M. Murray, and P. Rouchon, "Flat systems, equivalence, and trajectory generation," Technical Report, April 2003.

[5] B. Charlet, J. Lévine, and R. Marino, "Sufficient conditions for dynamic state feedback linearization," *SIAM J. Control and Optimization*, vol. 29, no. 1, pp. 38–57, 1991.

[6] P. Martin, "Contribution à l'étude des système différentiallement plats," Ph.D. Thesis, 1992, l'Ecole des Mines de Paris, France.

[7] ——, "Aircraft control using flatness," *Proceedings of the symposium on control, optimization, and supervision*, pp. 194–199, 1996.

[8] G. Pappas, "Avoiding saturation by trajectory reparameterization," *Proceedings of the 35th IEEE Conf. on Decision and Control*, pp. 76–81, 1996.

[9] S. Al-Hiddabi, "Comparison between trajectory tracking control and path following control: a linear system example," *Proceedings of the 23rd Chinese Control Conf. (International)*, 2004.

[10] S. Park, J. Deyst, and J. How, "A new nonlinear guidance logic for trajectory tracking," *Proceedings of the AIAA Conf. on Guidance, Navigation, and Control*, 2004.

[11] D. Gates, "Stable tracking to a straight path in the presence of wind in 3d," Personal correspondence, 2008.

[12] B. Etkin, *Dynamics of atmospheric flight.* Minneola, NY: Dover, 2000.

[13] H. Khalil, *Nonlinear systems.* Upper Saddle River, NJ: Prentice Hall, 2002.

[14] K. Åström and R. Murray, *Feedback systems: an introduction for scientists and engineers.* Princeton, NJ: Princeton University Press, 2008.